**ĐỀ KIỂM TRA SQL RETAKE ĐỢT 2**

**DÀNH CHO NHỮNG BẠN CHƯA THAM GIA ĐỢT 1**

**Cấu trúc** : Đề có tổng cộng **6 câu hỏi, chia làm 2 section**

* **DDL Testing section** có 3 câu kiểm tra các **lệnh DDL** : CREATE, ALTER, ADD, v.v
* **DML Testing section** có 3 câu kiểm tra các **lệnh DML** : SELECT, JOIN, GROUP BY, v.v

**Lưu ý**: Nội dung đề sẽ được trình bày **100% tiếng Anh** cho các câu hỏi

**I. DDL Testing Section (3 questions)**

**Quest 1**: Create database with name "**DEVER\_RETAKE\_SQL**". In this database, we will create **2 tables** for storing data with **the following requirements**

Table “**ClassRoom**” : To store [roomID, roomCode, roomName] with following constraints

* roomID : IDENTITY (Auto-Increasing) PRIMARY KEY
* roomCode : CHAR(7) NOT NULL
* roomName : VARCHAR(50) NOT NULL

Table "**Student**" : To store [studentID, studentCode, fullName, gender, dateOfBirth, roomID] with following constraints

* studentID : IDENTITY (Auto-Increasing) PRIMARY KEY
* studentCode : CHAR(8) NOT NULL
* fullName : VARCHAR(100) NOT NULL
* gender : BOOLEAN NOT NULL, **DEFAULT = true**
* roomCode : CHAR(7) NOT NULL

**Quest 2**: From the above database from **Quest 1**, using **ALTER** commands to add following constraint

Table " **ClassRoom**" :

* **roomCode** must be **UNIQUE**
* **roomCode** must start with ‘**SE**’, after ‘**SE**’ is the code of the room
  + The code of the room **must conform** the following format
    - Format = **\*\*@\*\*** , in that
      * **\*** : stand for **numeric** characters (ex: 1234567890)
      * **@** : stand for **alphabetic** character (ex: abcd …)
  + The roomCode examples that conforms the above format are “**Se16A08, SE15b22, sE19y65, …**”

Table “**Student**” :

* **studentCode** must be **UNIQUE**
* **studentCode** must start with ‘**DE**’, after ‘**DE**’ is the code of the student
  + The code of the student **must conform** the following format
    - Format = **\*\*\*\*\*\*** , in that
      * **\*** : stand for **numeric** characters (ex: 1234567890)
  + The **studentCode** examples that conforms the above format are “**de160505, dE170292, DE179999, …**”
* **roomCode** must be **FOREIGN KEY** and reference to **roomCode** in **ClassRoom table**

**Quest 3**: Insert following rows

-- ClassRoom Insert

1. INSERT INTO ClassRoom (roomCode, roomName) VALUES ('SE16A05', 'Lớp A05');

2. INSERT INTO ClassRoom (roomCode, roomName) VALUES ('Se16A09', 'Lớp A09');

3. INSERT INTO ClassRoom (roomCode, roomName) VALUES ('be14b04', 'Lớp B0B');

-- Student Insert

4. INSERT INTO Student (studentCode, fullName, roomCode) VALUES ('de170222', 'Nguyễn Văn A', 'SE16A05');

5. INSERT INTO Student (studentCode, fullName, roomCode) VALUES ('de170abc', 'Nguyễn Văn B', 'Se16A09');

6. INSERT INTO Student (studentCode, fullName, roomCode) VALUES ('de170111', 'Nguyễn Văn B', 'SE12A34');

**If your database is designed correctly**, the insert is success with **1. 2. 4.** insert statement

The insert statements **3. 5. 6.** are invalid

**II. DML Testing Section**

Using **dvdrental database** from the link: <https://www.postgresqltutorial.com/postgresql-getting-started/postgresql-sample-database/>

**Quest 1** (**Easy**): From the **Actor table**, find the actor who has appeared in the most films ever and display that actor information. If the result returns more than 1 actors, then selects the first actor in the result list

The extracted information **must include**

* actorID, fullname = firstname + lastname (**Ex: Bob Facet**)
* number of films that actor has appeared

**Quest 2** (**Medium**): From the **Payment table**, get the **latest payment record** in that table and extract the customer information who did the above latest payment. If the result returns more than 1 customer, then selects the first one in the result list  
  
The extracted customer information **must include**:

* customerID, email, customer city name, customer country name

**Quest 3** (**Hard**): From the **Film table**, find the film\_id with the highest revenue